```
                                     K 6

FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFFFFFFFFFFFFFF     OOOOOOOOO     RRRRRRRRRRRR     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFF              OOO         OOO  RRR         RRR  RRR         RRR        TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFFFFFFFFFFF     OOO         OOO  RRRRRRRRRRRR     RRRRRRRRRRRR           TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR   RRR        RRR   RRR              TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF              OOO         OOO  RRR      RRR     RRR      RRR           TTT        LLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
FFF                 OOOOOOOOO     RRR         RRR  RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
                                                M  1
**FILE**ID**FORUDFWU




          FFFFFFFFFF    OOOOOO    RRRRRRRR    UU      UU  DDDDDDDD    FFFFFFFFFF  WW      WW  UU      UU
          FFFFFFFFFF    OOOOOO    RRRRRRRR    UU      UU  DDDDDDDD    FFFFFFFFFF  WW      WW  UU      UU
          FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  UU      UU
          FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  UU      UU
          FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  UU      UU
          FF          OO      OO  RR      RR  UU      UU  DD      DD  FF          WW      WW  UU      UU
          FFFFFFFF    OO      OO  RRRRRRRR    UU      UU  DD      DD  FFFFFFFF    WW      WW  UU      UU
          FFFFFFFF    OO      OO  RRRRRRRR    UU      UU  DD      DD  FFFFFFFF    WW      WW  UU      UU
          FF          OO      OO  RR  RR      UU      UU  DD      DD  FF          WW  WW  WW  UU      UU
          FF          OO      OO  RR  RR      UU      UU  DD      DD  FF          WW  WW  WW  UU      UU
          FF          OO      OO  RR    RR    UU      UU  DD      DD  FF          WWWW  WWWW  UU      UU     ....
          FF          OO      OO  RR    RR    UU      UU  DD      DD  FF          WWWW  WWWW  UU      UU     ....
          FF            OOOOOO    RR      RR  UUUUUUUUUU  DDDDDDDD    FF          WW      WW  UUUUUUUUUU     ....
          FF            OOOOOO    RR      RR  UUUUUUUUUU  DDDDDDDD    FF          WW      WW  UUUUUUUUUU     ....


          LL            IIIIII      SSSSSSSS
          LL            IIIIII      SSSSSSSS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II        SSSSSS
          LL              II        SSSSSS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LLLLLLLLLL    IIIIII    SSSSSSSS
          LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE FOR$$UDF_WU (                               ! FORTRAN Write unformatted UDF
0               IDENT = '1-012'                      ! File: FORUDFWU.B32  Edit: JAW1012
0               ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:  FORTRAN Support Library - not user callable
1 !
1 ! ABSTRACT:
1 !
1 !       This module implements FORTRAN Write unformatted I/O
1 !       statements (sequential access - S, direct access - D,
1 !       at the User data Formatter level of
1 !       abstraction (UDF level is 2nd level). This module
1 !       calls the appropriate write record
1 !       routine at the record handling level of abstraction (REC
1 !       level is 3rd level) to write a record.
1 !
1 ! ENVIRONMENT:  User access mode; reentrant AST level or not.
1 !
1 ! AUTHOR:  Thomas N. Hastings;   CREATION DATE: 20-Feb-77
1 !
1 ! MODIFIED BY:
1 !       Thomas N. Hastings, 12-Mar-77: Version 01
1 ! 01    -  original
1 ! 0-4   - Removed parameters to record level routines JMT 17-OCT-77
1 ! 0-05  - Use FOR$K_abcmno05yz as E05TERNAL LITERALs.  TNH 27-Oct-77
1 ! 0-06  - Global register CCB.  JMT 8-Apr-78
1 ! 0-07  - Change REQUIRE files for VAX system build.  DGP 28-Apr-78
1 ! 0-08  - Use JSB linkages.  TNH 22-May-78
1 ! 0-09  - PIC dispatch tables.  TNH 7-June-78
1 ! 0-10  - Change file name to FORUDFWU.B32, and change the names of the
1 !         REQUIRE files similarly.  JBS 14-NOV-78
1 ! 1-001 - Update version number and copyright notice.  JBS 16-NOV-78
```

```
;     58          0058   1  ! 1-002 - Change REQUIRE file names from FOR... to OTS...  JBS 06-DEC-78
;     59          0059   1  ! 1-003 - Change references to I/O dipatch tables.  DGP 08-Dec-78
;     60          0060   1  ! 1-004 - Change dispatch tables references to longwords.  DGP 11-Dec-78
;     61          0061   1  ! 1-005 - Change ISB$A_BUF_PTR, BUF_BEG, BUF_END, BUF_HIGH to LUB. DGP 08-Jan-79
;     62          0062   1  ! 1-006 - Use 32-bit addresses for externals.  JBS 27-JAN-1979
;     63          0063   1  ! 1-007 - Use language-specific dispatch tables.  JBS 26-JUN-1979
;     64          0064   1  ! 1-008 - Use ISB dispatch offset symbols.  SBL 12-July-1979
;     65          0065   1  ! 1-009 - Signal an error if the record overflows and its not
;     66          0066   1  !         segmented.  V2 FT1 Report #61  SBL 18-Sept-1979
;     67          0067   1  ! 1-010 - Implement unbuffered transfers for single-element lists: if
;     68          0068   1  !         recordtype is not segmented and record size permits, set up
;     69          0069   1  !         RAB so record will be written directly from element, and do
;     70          0070   1  !         not copy element to buffer.  JAW 06-May-1981
;     71          0071   1  ! 1-011 - Continuation of 1-010.  Simplify tests in FOR$$UDF_WU1.
;     72          0072   1  !         JAW 02-Jun-1981
;     73          0073   1  ! 1-012 - Continuation of 1-010.  Recast test in FOR$$UDF_WU1 for better
;     74          0074   1  !         code.  Remove unnecessary resetting of flag in FOR$$UDF_WU9.
;     75          0075   1  !         JAW 06-Jun-1981
;     76          0076   1  !--
;     77          0077   1
;     78          0078   1  !<BLF/PAGE>
;    80         0079  1  !
;    81         0080  1  ! SWITCHES:
;    82         0081  1  !
;    83         0082  1
;    84         0083  1  SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
;    85         0084  1
;    86         0085  1  !
;    87         0086  1  ! LINKAGES:
;    88         0087  1  !
;    89         0088  1
;    90         0089  1  REQUIRE 'RTLIN:OTSLNK';                              ! Define all linkages
;    91         0518  1
;    92         0519  1  !
;    93         0520  1  ! TABLE OF CONTENTS:
;    94         0521  1  !
;    95         0522  1
;    96         0523  1  FORWARD ROUTINE
;    97         0524  1      FOR$$UDF_WU0 : JSB_UDF0 NOVALUE,                 ! initialization
;    98         0525  1      FOR$$UDF_WU1 : CALL_CCB NOVALUE,                 ! format one user I/O list element
;    99         0526  1      FOR$$UDF_WU9 : JSB_UDF9 NOVALUE;                 ! end of user I/O list - finish
;   100         0527  1
;   101         0528  1  !
;   102         0529  1  ! INCLUDE FILES:
;   103         0530  1  !
;   104         0531  1
;   105         0532  1  REQUIRE 'RTLML:FORERR';                              ! FORTRAN error codes
;   106         0600  1  REQUIRE 'RTLML:OTSISB';                              ! I/O statement block (ISB) offsets
;   107         0768  1
;   108         0769  1  REQUIRE 'RTLML:OTSLUB';                              ! Logical unit block (LUB)
;   109         0909  1
;   110         0910  1  REQUIRE 'RTLIN:OTSMAC';                              ! Macros
;   111         1104  1
;   112         1105  1  REQUIRE 'RTLIN:RTLPSECT';                            ! Define DELCARE_PSECTS macro
;   113         1200  1
;   114         1201  1  REQUIRE 'RTLIN:RTLODBG';                             ! RTL debugging macros
;   115         1310  1
;   116         1311  1  LIBRARY 'RTLSTARLE';
;   117         1312  1
;   118         1313  1  !
;   119         1314  1  ! MACROS:
;   120         1315  1  !
;   121         1316  1  !
;   122         1317  1  ! EQUATED SYMBOLS:
;   123         1318  1  !
;   124         1319  1                                                       ! all other FORTRAN data type codes are larger valued
;   125         1320  1  !
;   126         1321  1  ! PSECT DECLARATIONS:
;   127         1322  1  !
;   128         1323  1  DECLARE_PSECTS (FOR);                                ! declare PSECTs for FOR$ facility
;   129         1324  1  !
;   130         1325  1  ! OWN STORAGE:
;   131         1326  1  !
;   132         1327  1  !       None
;   133         1328  1  !
;   134         1329  1  ! EXTERNAL REFERENCES:
;   135         1330  1  !
;   136         1331  1
;  137          1332  1 EXTERNAL
;  138          1333  1     FOR$$AA_REC_PR0 : VECTOR,                     ! PIC array of record processor
;  139          1334  1                                                   ! procedure-initializations in REC
;  140          1335  1                                                   ! level of abstraction. Indexed by
;  141          1336  1                                                   ! I/O statement type (ISB$B_STTM_TYPE)
;  142          1337  1     FOR$$AA_REC_PR1 : VECTOR,                     ! PIC array of record processor procedures
;  143          1338  1                                                   ! Write a record in REC level of
;  144          1339  1                                                   ! abstraction. Indexed by I/O statement
;  145          1340  1                                                   ! type (ISB$B_STTM_TYPE)
;  146          1341  1     FOR$$AA_REC_PR9 : VECTOR;                     ! PIC array of record processor procedures
;  147          1342  1
;  148          1343  1                                                   ! Write last record in REC level of
;  149          1344  1                                                   ! abstraction. Indexed by I/O
;  150          1345  1                                                   ! statement type (ISB$B_STTM_TYPE)
;  151          1346  1
;  152          1347  1 EXTERNAL ROUTINE
;  153          1348  1     FOR$$SIGNAL_STO : NOVALUE;                    ! Signal fatal error
;  1 GLOBAL ROUTINE FOR$$UDF_WUO                        ! Write unformatted UDF initialization
;  1     : JSB_UDFO NOVALUE =
;  1
;  1 !++
;  1 ! FUNCTIONAL DESCRIPTION:
;  1 !
;  1 ! Initialize Write unformatted User data formatter (UDF)
;  1 !
;  1 ! CALLING SEQUENCE:
;  1 !
;  1 !       JSB FOR$$UDF_WUO ()
;  1 !
;  1 ! FORMAL PARAMETERS:
;  1 !
;  1 !       None.
;  1 !
;  1 ! IMPLICIT INPUTS:
;  1 !
;  1 !       CCB                     Pointer to current logical unit block
;  1 !       ISB$B_STTM_TYPE         I/O statement type code - set by
;  1 !                               each I/O statement initialization
;  1 !
;  1 ! IMPLICIT OUTPUTS:
;  1 !
;  1 !       LUB$A_BUF_PTR           Adr. of next byte of output
;  1 !                               data buffer
;  1 !       LUB$A_BUF_END           Adr. of end+1 byte position of output buffer
;  1 !
;  1 ! ROUTINE VALUE:
;  1 ! COMPLETION CODES:
;  1 !
;  1 !       NONE
;  1 !
;  1 ! SIDE EFFECTS:
;  1 !
;  1 !       NONE
;  1 !
;  1 !--
;  1
;  2     BEGIN
;  2
;  2     EXTERNAL REGISTER
;  2         CCB : REF BLOCK [, BYTE];
;  2
;  2     !+
;  2     ! Initialize Record processing level of abstraction.
;  2     ! Set pointer to current (LUB$A_BUF_PTR)  and last+1
;  2     ! (LUB$A_BUF_END) character position for user data in
;  2     ! output buffer
;  2     !-
;  2
;  2     JSB_RECO (FOR$$AA_REC_PRO + .FOR$$AA_REC_PRO [.CCB [ISB$B_STTM_TYPE] -
;  2                                            ISB$K_FORSTTYLO + 1]);
;  2
;  2     RETURN;
;  1     END;                                   ! End of FOR$$UDF_WUO routine
```

```
                                                       .TITLE  FOR$$UDF_WU
                                                       .IDENT  \1-012\

                                                       .EXTRN  FOR$$AA_REC_PR0
                                                       .EXTRN  FOR$$AA_REC_PR1
                                                       .EXTRN  FOR$$AA_REC_PR9
                                                       .EXTRN  FOR$$SIGNAL_STO

                                                       .PSECT  _FOR$CODE,NOWRT,  SHR,  PIC,2

                           50      FF71   CB  9A 00000 FOR$$UDF_WU0::
                                                       MOVZBL  -143(CCB), R0                                 ; 1401
                           50 00000000G0040   D0 00005 MOVL    FOR$$AA_REC_PR0[R0], R0                       ; 1400
                              00000000G0040   17 0000D JMP     FOR$$AA_REC_PR0[R0]                           ;

; Routine Size:  20 bytes,    Routine Base:  _FOR$CODE + 0000


;  211          1405  1
;   213         1406    1 GLOBAL ROUTINE FOR$$UDF_WU1 (                         ! Copy one user output element
;   214         1407    1         ELEM_TYPE,                                    ! Type code of user I/O list element
;   215         1408    1         ELEM_SIZE,                                    ! No. of addressable units in element
;   216         1409    1         ELEM_ADR)                                     ! Adr. of element
;   217         1410    1     : CALL_CCB NOVALUE =
;   218         1411    1
;   219         1412    1 !++
;   220         1413    1 ! FUNCTIONAL DESCRIPTION:
;   221         1414    1 !
;   222         1415    1 !       FOR$$UDF_WU1 copies a single user I/O list element
;   223         1416    1 !       and places it in the current output buffer.
;   224         1417    1 !       If the entire user element will not fit in the buffer,
;   225         1418    1 !       only part if copied and the buffer is output by calling
;   226         1419    1 !       the proper record level (3rd level of abstraction).
;   227         1420    1 !
;   228         1421    1 !       If an unbuffered transfer is possible, no data is copied, and
;   229         1422    1 !       the RAB is set up to point to the element itself.
;   230         1423    1 !
;   231         1424    1 !       FOR$$UDF_WU is the same for all access modes.
;   232         1425    1 !
;   233         1426    1 ! CALLING SEQUENCE:
;   234         1427    1 !
;   235         1428    1 !       CALL FOR$$UDF_WU1 (elem_type.rlu.v, elem_size.rlu.v, elem_adr.rx.r)
;   236         1429    1 !
;   237         1430    1 ! FORMAL PARAMETERS:
;   238         1431    1 !
;   239         1432    1 !       ELEM_TYPE.rlu.v         Type code of user I/O list
;   240         1433    1 !                               element. Form: ELEM_TYPE_x
;   241         1434    1 !                               x = B,W,L,WU,LU,F,D,FC,T,G,H,DC or GC.
;   242         1435    1 !       ELEM_SIZE.rlu.v         Size of user I/O list element
;   243         1436    1 !                               in addressable machine units
;   244         1437    1 !       ELEM_ADR.rx.r           Adr. of user I/O list element
;   245         1438    1 !                               x = b, w, l, wu, lu, f, d, fc, t,
;   246         1439    1 !                               g, h, dc or gc.
;   247         1440    1 !
;   248         1441    1 ! IMPLICIT INPUTS:
;   249         1442    1 !
;   250         1443    1 !       OTS$$A_CUR_LUB          Pointer to current logical unit block
;   251         1444    1 !                               (LUB). Used to setup base pointer ISB
;   252         1445    1 !                               to current I/O statement block
;   253         1446    1 !       ISB$B_STTM_TYPE         I/O statement type code - set by each
;   254         1447    1 !                               I/O statement initialization
;   255         1448    1 !       ISB$V_SNGL_ELEM         Flag indicating that this element is the
;   256         1449    1 !                               only element in the current I/O list.
;   257         1450    1 !
;   258         1451    1 ! The following ISB locations are set only by previous calls to
;   259         1452    1 ! FOR$$UDF_WU{0,1}, i.e., are effectively OWN.
;   260         1453    1 !
;   261         1454    1 !       LUB$A_BUF_PTR           Pointer to next char. position
;   262         1455    1 !                               in user data part of output buffer
;   263         1456    1 !       LUB$A_BUF_END           Adr. of last+1 byte position of output buffer
;   264         1457    1 !
;   265         1458    1 ! IMPLICIT OUTPUTS:
;   266         1459    1 !
;   267         1460    1 !       ISB$V_SNGL_ELEM         This flag is cleared if an unbuffered
;   268         1461    1 !                               transfer is not possible.
;   269         1462    1 !
```

```
1 ! The following ISB locations are set only by previous calls
1 ! to FOR$$UDF_WU{0,1}, i.e., are effectively OWN.
1 !
1 !      LUB$A_BUF_PTR              Pointer to next char. position
1 !                                 in user data part of output buffer
1 !
1 ! FUNCTIONAL VALUE:
1 !
1 !      NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !      SIGNAL_STOPs FOR$_OUTSTAOVE (Output statement overflows record)
1 !--
1
2     BEGIN
2
2     EXTERNAL REGISTER
2         CCB : REF BLOCK [, BYTE];
2
2     !+
2     ! If this is a single-element list, check to see if the conditions
2     ! for an unbuffered transfer are met.  If so, set RAB$L_RBF and
2     ! RAB$W_RSZ to point directly to the element, and return leaving
2     ! ISB$V_SNGL_ELEM set as an indication to REC level not to change
2     ! them.  Otherwise clear ISB$V_SNGL_ELEM and proceed normally.
2     !-
2
2     IF .CCB [ISB$V_SNGL_ELEM]
2     THEN
2         IF NOT .CCB [LUB$V_SEGMENTED] AND
3             (IF .CCB [LUB$V_FIXED] THEN .ELEM_SIZE EQLU .CCB [LUB$W_RBUF_SIZE]
3             ELSE .ELEM_SIZE LSSU .CCB [LUB$W_RBUF_SIZE])
2         THEN
3             BEGIN
3                 CCB [RAB$L_RBF] = .ELEM_ADR;
3                 CCB [RAB$W_RSZ] = .ELEM_SIZE;
3                 RETURN;
3             END
2         ELSE
2             CCB [ISB$V_SNGL_ELEM] = 0;
2
3     BEGIN
3
3     LOCAL
3         TMP_ELEM_SIZE,                    ! temp no. ob bytes left in user element to copy
3         TMP_ELEM_ADR,                     ! temp adr. of rest of user element to copy
3         TMP_DIFF;                                   ! temp. no. of bytes to move each time in loop
3
3     !+
3     ! If the record will overflow, write as much as will fit.  If this
3     ! is done, and the file is not SEGMENTED, then error FOR$_OUTSTAOVE.
3     ! If segmented, continue writing records until overflow is removed.
3     !-
3
3     TMP_ELEM_SIZE = .ELEM_SIZE;
3     TMP_ELEM_ADR = .ELEM_ADR;
```

```
;  327          1520  3
;  328          1521  3       WHILE .CCB [LUB$A_BUF_PTR] + .TMP_ELEM_SIZE GTRA .CCB [LUB$A_BUF_END] DO
;  329          1522  4           BEGIN
;  330          1523  4           TMP_DIFF = .CCB [LUB$A_BUF_END] - .CCB [LUB$A_BUF_PTR];
;  331          1524  4           CCB [LUB$A_BUF_PTR] = CH$MOVE (.TMP_DIFF, .TMP_ELEM_ADR, .CCB [LUB$A_BUF_PTR]);
;  332          1525  4           TMP_ELEM_SIZE = .TMP_ELEM_SIZE - .TMP_DIFF;
;  333          1526  4           TMP_ELEM_ADR = .TMP_ELEM_ADR + .TMP_DIFF;
;  334          1527  4
;  335          1528  4           JSB_REC1 (FOR$$AA_REC_PR1 + .FOR$$AA_REC_PR1 [.CCB [ISB$B_STTM_TYPE] -
;  336          1529  4                                                   ISB$K_FORSTTYLO + 1]);
;  337          1530  4           IF NOT .CCB [LUB$V_SEGMENTED]
;  338          1531  4           THEN
;  339          1532  4               FOR$$SIGNAL_STO (FOR$K_OUTSTAOVE);
;  340          1533  3           END;
;  341          1534  3
;  342          1535  3           !+
;  343          1536  3           ! Copy the remainder of the user element that will fit into the output buffer
;  344          1537  3           ! Update buffer pointer (LUB$A_BUF_PTR) to point to last byte+1 moved.
;  345          1538  3           !-
;  346          1539  3
;  347          1540  3           CCB [LUB$A_BUF_PTR] = CH$MOVE (.TMP_ELEM_SIZE, .TMP_ELEM_ADR, .CCB [LUB$A_BUF_PTR]);
;  348          1541  3           RETURN;                                   ! Return from FOR$$UDF_WF1 routine
;  349          1542  2       END;
;  350          1543  1 END;                                 !          End of FOR$$UDF_WU1




                                          03FC 00000           .ENTRY  FOR$$UDF_WU1, Save R2,R3,R4,R5,R6,R7,R8,R9   ; 1406
                     59 0000000G    00     9E 00002           MOVAB   FOR$$AA_REC_PR1, R9                          ;
              2D      97    AB      04     E1 00009           BBC     #4, -105(CCB), 4$                            ; 1491
              24      FD    AB      03     E0 0000E           BBS     #3, -3(CCB), 3$                              ; 1493
              0B      FD    AB      02     E1 00013           BBC     #2, -3(CCB), 1$                              ; 1494
  08    AC    D2    AB      10      00     ED 00018           CMPZV   #0, #16, -46(CCB), ELEM_SIZE                 ;
                            16      12 0001F           BNEQ    3$                                           ;
                            09      11 00021           BRB     2$                                           ;
  08    AC    D2    AB      10      00     ED 00023 1$:       CMPZV   #0, #16, -46(CCB), ELEM_SIZE                 ; 1495
                                    0B     1B 0002A           BLEQU   3$                                           ;
                      28    AB    0C    AC     D0 0002C 2$:       MOVL    ELEM_ADR, 40(CCB)                            ; 1498
                      22    AB    08    AC     B0 00031           MOVW    ELEM_SIZE, 34(CCB)                           ; 1499
                                           04 00036           RET                                                  ; 1497
                      97    AB      10     8A 00037 3$:       BICB2   #16, -105(CCB)                               ; 1503
                            58    08    AC     D0 0003B 4$:       MOVL    ELEM_SIZE, TMP_ELEM_SIZE                     ; 1518
                            57    0C    AC     D0 0003F           MOVL    ELEM_ADR, TMP_ELEM_ADR                       ; 1519
              50            58    B0    AB     C1 00043 5$:       ADDL3   -80(CCB), TMP_ELEM_SIZE, R0                  ; 1521
                      B4    AB      50     D1 00048           CMPL    R0, -76(CCB)                                 ;
                                    33     1B 0004C           BLEQU   6$                                           ;
              56      B4    AB    B0    AB     C3 0004E           SUBL3   -80(CCB), -76(CCB), TMP_DIFF                 ; 1523
        B0    BB            67      56     28 00054           MOVC3   TMP_DIFF, (TMP_ELEM_ADR), @-80(CCB)          ; 1524
                      B0    AB      53     D0 00059           MOVL    R3, -80(CCB)                                 ;
                            58      56     C2 0005D           SUBL2   TMP_DIFF, TMP_ELEM_SIZE                      ; 1525
                            57      56     C0 00060           ADDL2   TMP_DIFF, TMP_ELEM_ADR                       ; 1526
                            50  FF71    CB     9A 00063           MOVZBL  -143(CCB), R0                                ; 1529
                            50      6940   D0 00068           MOVL    FOR$$AA_REC_PR1[R0], R0                      ; 1528
                                    6940   16 0006C           JSB     FOR$$AA_REC_PR1[R0]                          ;
              CF      FD    AB      03     E0 0006F           BBS     #3, -3(CCB), 5$                              ; 1530
```

```
                              7E        42    8F  9A 00074        MOVZBL  #66, -(SP)                                ; 1532
                  00000000G   00              01  FB 00078        CALLS   #1, FOR$$SIGNAL_STO                       ;
                                              C2  11 0007F        BRB     5$                                        ; 1521
             B0   BB          67              58  28 00081  6$:   MOVC3   TMP_ELEM_SIZE, (TMP_ELEM_ADR), @-80(CCB)  ; 1540
                        B0    AB              53  D0 00086        MOVL    R3, -80(CCB)                              ;
                                                  04 0008A        RET                                               ; 1543

; Routine Size:  139 bytes,    Routine Base:  _FOR$CODE + 0014


;  351          1544  1
;  353        1545  1 GLOBAL ROUTINE FOR$$UDF_WU9                         ! unformatted output - end of I/O list call
;  354        1546  1     : JSB_UDF9 NOVALUE =
;  355        1547  1
;  356        1548  1 !++
;  357        1549  1 ! FUNCTIONAL DESCRIPTION:
;  358        1550  1 !
;  359        1551  1 !       FOR$$UDF_WU9 performs end of I/O list output formatting.
;  360        1552  1 !
;  361        1553  1 ! CALLING SEQUENCE:
;  362        1554  1 !
;  363        1555  1 !       JSB FOR$$UDF_WU9 ()
;  364        1556  1 !
;  365        1557  1 ! FORMAL PARAMETERS:
;  366        1558  1 !
;  367        1559  1 !       NONE
;  368        1560  1 !
;  369        1561  1 ! IMPLICIT INPUTS:
;  370        1562  1 !
;  371        1563  1 !       See FOR$$UDF_WU1
;  372        1564  1 !
;  373        1565  1 !
;  374        1566  1 ! IMPLICIT OUTPUTS:
;  375        1567  1 !
;  376        1568  1 !       See FOR$$UDF_WU1
;  377        1569  1 !
;  378        1570  1 ! FUNCTION VALUE:
;  379        1571  1 !
;  380        1572  1 !       NONE
;  381        1573  1 !
;  382        1574  1 ! SIDE EFFECTS:
;  383        1575  1 !
;  384        1576  1 !       See FOR$$UDF_WU1
;  385        1577  1 !--
;  386        1578  1
;  387        1579  2     BEGIN
;  388        1580  2
;  389        1581  2     EXTERNAL REGISTER
;  390        1582  2         CCB : REF BLOCK [, BYTE];
;  391        1583  2
;  392        1584  2     !+
;  393        1585  2     ! Call record level of abstraction to output buffer from beginning up to but not including LUB$A_BUF_PTR
;  394        1586  2     !-
;  395        1587  2
;  396        1588  2     JSB_REC9 (FOR$$AA_REC_PR9 + .FOR$$AA_REC_PR9 [.CCB [ISB$B_STTM_TYPE] -
;  397        1589  2                                                 ISB$K_FORSTTYLO + 1]);
;  398        1590  2
;  399        1591  2     RETURN;
;  400        1592  1     END;                                        ! End of FOR$$UDF_WU9 Routine



                              50      FF71   CB  9A 00000 FOR$$UDF_WU9::
                                                               MOVZBL  -143(CCB), R0                          ; 1589
                              50 00000000G0040  D0 00005      MOVL    FOR$$AA_REC_PR9[R0], R0                ; 1588
                                 00000000G0040  17 0000D      JMP     FOR$$AA_REC_PR9[R0]                    ;
```

```
; Routine Size:  20 bytes,    Routine Base:  _FOR$CODE + 009F

;  401          1593  1
;  402          1594  1 END                                                   ! End of FOR$$UDF_WU Module
;  403          1595  1
;  404          1596  0 ELUDOM



;                            PSECT SUMMARY
;
;     Name                    Bytes                    Attributes
;
;  _FOR$CODE                             179  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                       Library Statistics
;
;                                         -------- Symbols --------        Pages       Processing
;     File                                Total    Loaded   Percent       Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1       9776        2         0          581         00:01.1



;                             COMMAND QUALIFIERS

;     BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:FORUDFWU/OBJ=OBJ$:FORUDFWU MSRC$:FORUDFWU/UPDATE=(ENH$:FORUDFWU)

; Size:          179 code + 0 data bytes
; Run Time:         00:13.3
; Elapsed Time:     00:40.0
; Lines/CPU Min:      7194
; Lexemes/CPU-Min: 37523
; Memory Used:  159 pages
; Compilation Complete
```

0185 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY